

CURSO: Sistemas de Informação

DISCIPLINA: Arquitetura de Computadores

PROFESSOR: Gabriel Novy

VALOR: 15 pontos

DATA: 10/06/2024

TRABALHO 1

A arquitetura MIPS é uma arquitetura de computadores do tipo RISC utilizada em alguns eletrônicos, com os consoles PlayStation e PlayStation 2. Essa arquitetura possui as principais características de ser simples, de fácil entendimento e estudada nos cursos da área de Computação. Esta é a arquitetura que tem sido utilizada para estudo de caso na disciplina de Arquitetura de Computadores.

Este trabalho tem como objetivo levar o aluno a conhecer e entender o funcionamento das instruções dessa arquitetura. Para isso, deverá ser feito um **programa que deverá ler um arquivo contendo os código MIPS binários da instrução e gerar um arquivo de saída contendo os mnemônicos correspondentes**. O arquivo de entrada conterá N instruções escritas em binário (sendo $N > 0$). O arquivo de saída deverá conter as mesmas N instruções, na mesma ordem, mas escritas com os seus respectivos mnemônicos. Cada uma das instruções do arquivo de entrada estará em uma linha, e o mesmo deve acontecer com o arquivo de saída. A seguir, temos um exemplo de arquivo de entrada e a saída esperada:

ENTRADA	SAÍDA
TESTE-01.txt	TESTE-01-RESULTADO.txt
10001101001010000000010010110000	lw \$t0, 1200(\$t1)
00000010010010000100000000100000	add \$t0, \$s2, \$t0
10101101001010000000010010110000	sw \$t0, 1200(\$t1)

Os arquivos podem conter qualquer instrução das descritas no anexo I. Para maiores informações sobre as instruções, procure na internet ou o professor em seu horário de atendimento (quintas-feiras, das 13:00 às 14:00). É necessário agendar com antecedência.

Para os testes a serem realizados no dia da apresentação, o programa deverá ler, em sequência, um total de 10 arquivos de entrada que **estarão na raiz de um pendrive** e que serão nomeados seguindo o formato “TESTE-XX.txt”, sendo XX o número do arquivo de teste que irá de 01 a 10. Para cada arquivo de entrada, o programa deverá produzir um arquivo de saída contendo a resposta e salvar esses arquivos no mesmo diretório dos arquivos de entrada. Além disso, o arquivo de saída deverá ter o mesmo nome do arquivo de entrada acrescido de “-RESULTADO.txt”. Ou seja, se o arquivo de entrada tiver o nome “TESTE-01.txt”, o arquivo de saída deverá conter o nome “TESTE-01-RESULTADO.txt”. Abaixo segue uma imagem de como estará a pasta de arquivos de teste antes da execução do seu programa e como é esperado que ela esteja após a execução do seu programa.

ANTES DA EXECUÇÃO DO PROGRAMA		APÓS A EXECUÇÃO DO PROGRAMA	
Nome	Data de mo...	Nome	Data de mo...
TESTE-01.txt	12/08/2024 ...	TESTE-10.txt	12/08/2024 ...
TESTE-02.txt	12/08/2024 ...	TESTE-09.txt	12/08/2024 ...
TESTE-03.txt	12/08/2024 ...	TESTE-08.txt	12/08/2024 ...
TESTE-04.txt	12/08/2024 ...	TESTE-07.txt	12/08/2024 ...
TESTE-05.txt	12/08/2024 ...	TESTE-06.txt	12/08/2024 ...
TESTE-06.txt	12/08/2024 ...	TESTE-05.txt	12/08/2024 ...
TESTE-07.txt	12/08/2024 ...	TESTE-04.txt	12/08/2024 ...
TESTE-08.txt	12/08/2024 ...	TESTE-03.txt	12/08/2024 ...
TESTE-09.txt	12/08/2024 ...	TESTE-02.txt	12/08/2024 ...
TESTE-10.txt	12/08/2024 ...	TESTE-01.txt	12/08/2024 ...
		TESTE-01-RESULTADO.txt	12/08/2024 ...
		TESTE-02-RESULTADO.txt	12/08/2024 ...
		TESTE-03-RESULTADO.txt	12/08/2024 ...
		TESTE-04-RESULTADO.txt	12/08/2024 ...
		TESTE-05-RESULTADO.txt	12/08/2024 ...
		TESTE-06-RESULTADO.txt	12/08/2024 ...
		TESTE-07-RESULTADO.txt	12/08/2024 ...
		TESTE-08-RESULTADO.txt	12/08/2024 ...
		TESTE-09-RESULTADO.txt	12/08/2024 ...
		TESTE-10-RESULTADO.txt	12/08/2024 ...

O trabalho será desenvolvido em dupla e o software poderá ser implementado utilizando a linguagem que a dupla se sentir mais confortável. No entanto, **certifique-se de que os laboratórios possuem instalados os recursos necessários para a execução do programa. Se a dupla preferir, poderá utilizar notebook pessoal para a apresentação.**

A entrega ocorrerá exclusivamente pelo SUAP até o dia **10 de junho de 2025, às 23:59**. O trabalho, contendo **APENAS O CÓDIGO FONTE**, deverá ser entregue em um arquivo compactado (ZIP ou RAR) contendo apenas o nome e sobrenome dos dois integrantes da dupla. Exemplo: "Gabriel Novy e Cristiane Targa.zip". **ATENÇÃO:** Não é "Trabalho II - Fulano e Ciclano.zip" e nem "Trabalho Arquitetura.rar". Se atente às instruções.

O trabalho tem valor de 15 pontos e a avaliação seguirá a seguintes distribuição:

- 1 ponto para cada um dos 10 arquivos de teste: se o arquivo de saída corresponder ao esperado é acerto e ganha 1 ponto. Se o arquivo de saída não corresponder ao esperado, por qualquer motivo que seja, é 0 (zero). Então fique atento e siga as regras, inclusive no nome do arquivo de saída.
- 2,5 pontos para perguntas que irei fazer durante a execução dos testes.
- 2,5 pontos para o código.

A **apresentação será no dia 11 de junho de 2025 no laboratório 101** e os dois integrantes da dupla deverão estar presentes. Cada dupla terá, no máximo, 5 minutos para a sua apresentação, que seguirá uma escala com o horário definido de apresentação. As duplas deverão enviar e-mail para gabriel.novy@ifmg.edu.br informando o nome dos integrantes. A escala será definida pela ordem inversa de envio dos e-mails. Quem enviar primeiro ficará por último na escala de apresentação. A escala será disponibilizada neste mesmo arquivo posteriormente. Apenas serão aceitas apresentações posteriores nos casos previstos na legislação.

INTEGRANTES DA DUPLA	HORÁRIO DE APRESENTAÇÃO
Tiago Silva	7:10
	7:15
Maurício Pessoa e Rodrigo Cardoso	7:20
Athos Alves	7:25
Rodrigo Albano e Marcella Del Rio	7:30
Augusto Henrique e Matheus Carneiro	7:35
João Pedro Aguiar e Victor Lopes	7:40
Kawan Luis e Bruna Gonçalves	7:45
INTERVALO	7:50 - 8:00
Raissa Roberta e Julia Cavalcante	8:00
Douglas Henrique e João Victor Marçal	8:05
Gabriel Nunes e Lucas Epifanio	8:10
Tiago Mol e Mateus Athos	8:15
Maria Paula e Caio Caldas	8:20
Thiago Henrique e Fabio Vieira	8:25
Lucas Vinicius Poeiras e Robert Barbosa	8:30
Gustavo de Oliveira e Rafael Chapman	8:35

ANEXO I

FORMATO DE INSTRUÇÕES

Type	-31- format (bits) -0-					
R	opcode (6)	rs (5)	rt (5)	rd (5)	shamt (5)	funct (6)
I	opcode (6)	rs (5)	rt (5)	immediate (16)		
J	opcode (6)	address (26)				

INSTRUÇÕES LOAD E STORE

Instruction name	Mnemonic	Format	Encoding				EXEMPLO
Load Byte	LB	I	32 ₁₀	rs	rt	offset	lb \$s1, 200(\$s2)
Load Halfword	LH	I	33 ₁₀	rs	rt	offset	lh \$s1, 200(\$s2)
Load Word	LW	I	35 ₁₀	rs	rt	offset	lw \$s1, 200(\$s2)
Store Byte	SB	I	40 ₁₀	rs	rt	offset	sb \$s1, 200(\$s2)
Store Halfword	SH	I	41 ₁₀	rs	rt	offset	sh \$s1, 200(\$s2)
Store Word	SW	I	43 ₁₀	rs	rt	offset	sw \$s1, 200(\$s2)

INSTRUÇÕES LÓGICAS E ARITMÉTICAS

Instruction name	Mnemonic	Format	Encoding						EXEMPLO
Add	ADD	R	0 ₁₀	rs	rt	rd	0 ₁₀	32 ₁₀	add \$s1, \$s2, \$s3
Subtract	SUB	R	0 ₁₀	rs	rt	rd	0 ₁₀	34 ₁₀	sub \$s1, \$s2, \$s3
And	AND	R	0 ₁₀	rs	rt	rd	0 ₁₀	36 ₁₀	and \$s1, \$s2, \$s3
Or	OR	R	0 ₁₀	rs	rt	rd	0 ₁₀	37 ₁₀	or \$s1, \$s2, \$s3
Exclusive Or	XOR	R	0 ₁₀	rs	rt	rd	0 ₁₀	38 ₁₀	xor \$s1, \$s2, \$s3
Add Immediate	ADDI	I	8 ₁₀	rs	rd	immediate			addi \$s1, \$s2, 245
And Immediate	ANDI	I	12 ₁₀	\$s	\$d	immediate			andi \$s1, \$s2, 245
Or Immediate	ORI	I	13 ₁₀	\$s	\$d	immediate			ori \$s1, \$s2, 245
Exclusive Or Immediate	XORI	I	14 ₁₀	\$s	\$d	immediate			xori \$s1, \$s2, 245
Load Upper Immediate	LUI	I	15 ₁₀	0 ₁₀	\$d	immediate			liu \$s1, \$s2, 245

INSTRUÇÕES DE DESLOCAMENTO DE BITS

Instruction name	Mnemonic	Format	Encoding						EXEMPLO
Shift Left Logical	SLL	R	0 ₁₀	0 ₁₀	rt	rd	ra	0 ₁₀	sll \$s1, \$s2, 12
Shift Right Logical	SRL	R	0 ₁₀	0 ₁₀	rt	rd	sa	2 ₁₀	srl \$s1, \$s2, 12

INSTRUÇÕES DE DESVIO

Instruction name	Mnemonic	Format	Encoding						EXEMPLO
Jump Register	JR	R	0 ₁₀	rs	0 ₁₀	0 ₁₀	0 ₁₀	8 ₁₀	jr \$s1
Jump	J	J	2 ₁₀	instr_index					j 1024
Branch on Equal	BEQ	I	4 ₁₀	rs	rt	offset			beq \$s1, \$s2, 300
Branch on Not Equal	BNE	I	5 ₁₀	rs	rt	offset			bne \$s1, \$s2, 300
Branch on Less Than or Equal to Zero	BLEZ	I	6 ₁₀	rs	0 ₁₀	offset			blez \$s1, 300
Branch on Greater Than Zero	BGTZ	I	7 ₁₀	rs	0 ₁₀	offset			bgtz \$s1, 300